



Uniwersytet Gdański  
Wydział Matematyki, Fizyki i Informatyki  
Instytut Informatyki

# System wirtualnej uczelni

Marian Buzak



Projekt z przedmiotu bazy danych na kierunku informatyka profil ogólnoakademicki na Uniwersytecie Gdańskim.

Gdańsk 25 maja 2020

## Spis treści

<b>1</b>	<b>Wprowadzenie</b>	<b>2</b>
<b>2</b>	<b>Opis projektu</b>	<b>2</b>
2.1	Potencjalne grupy użytkowników . . . . .	2
2.2	Wymagania funkcjonalne . . . . .	3
2.3	Wymagania niefunkcjonalne . . . . .	3
2.4	Diagram związków encji . . . . .	3
<b>3</b>	<b>Przykłady realizacji bazy danych</b>	<b>3</b>
3.1	Przykłady zawartości najważniejszych tabel . . . . .	3
3.2	Przykłady kilku zapytań i ich wyników . . . . .	5

# 1 Wprowadzenie

Baza danych przeznaczona jest dla władz uczelni. Dzięki niej można znaleźć potrzebne informacje zarówno o studentach jak i o nauczycielach. Finanse są istotne - dlatego też można sprawdzić jakie opłaty zostały dokonane przez studentów, ale też, którym studentom przypada premia finansowa.

Słownik wyrazów:



- System bazodanowy - **serwer służący** do obsługi bazy danych
- Tabela - wydzielony logicznie zbiór danych składający się z wierszy podzielonych na kolumny
- Krotka - **struktura danych** będąca odzwierciedleniem uporządkowanego ciągu wartości
- Redundancja - zbędne powtarzanie informacji
- Normalizacja - proces uporządkowania danych m.in. podział danych na tabele, ustanowienie relacji pomiędzy tabelami, usunięcia redundancji
- Klucz podstawowy - minimalny zestaw atrybutów relacji, który przyjmuje wartości niepowtarzalne oraz niepuste
- Klucz obcy - kombinacja atrybutów tabeli, które wyrażają się w dwóch lub większej liczbie **relacji**



## 2 Opis projektu

W przeszłości pewna uczelnia gromadziła informacje w skoroszytach. Niestety zdarzyło się, że w budynku powstał pożar, który zniszczył te dane. Dlatego też uczelnia potrzebuje bazy danych, dzięki której będzie można wyszukać informacje o studentach. Ta uczelnia potrzebuje takiego rozwiązania, ponieważ gdyby nie taka baza, to mogłoby doprowadzić do ponownej utraty informacji.

### 2.1 Potencjalne grupy użytkowników

Potencjalni użytkownicy bazy danych:

- Administrator – główny zarządca bazy danych, posiada pełen dostęp do bazy danych
- Student - rodzaj użytkownika mający najbardziej ograniczony dostęp do danych - dostęp do swoich danych personalnych
- Nauczyciel - użytkownik mający dostęp do danych swoich oraz studentów
- Dziekan - ma dostęp do m.in. składanych opłat, informacji o nauczycielach oraz studentach

## 2.2 Wymagania funkcjonalne

W bazie danych znajdują się informacje o wydziałach, kierunkach, akademikach. **Ważnym** aspektem jest przechowywanie danych osobowych, dlatego są tutaj dane o studentach oraz pracownikach. Każdy student jest przypisany do grup ćwiczeniowych. Zajęcia są zorganizowane - można odczytać jakiego dnia odbyły się zajęcia z danej grupy. Dzięki bazie danych można znaleźć adresy poszczególnych osób. Można również znaleźć informacje na temat opłat studenta oraz historii prac dyplomowych. Dzięki temu można utworzyć ich ranking.



## 2.3 Wymagania niefunkcjonalne

**Projekt zgodny z MySQL**, zamieszczony na **serwerze MySQL v5.7.29-0ubuntu0.18.04.1** (Ubuntu).

Istotną wadą MySQL jest jego brak integralności z językiem SQL. Dlatego przełączenie na inny SZBD (system zarządzania bazą danych) nie jest łatwe. Również problematyczne może być operowanie tekstami, ponieważ MySQL nie posiada niektórych funkcji np. wyszukiwanie pełnotekstowe. Natomiast ważnymi zaletami MySQL jest wysoki poziom bezpieczeństwa oraz jego prosta instalacja i obsługa.



## 2.4 Diagram związków encji

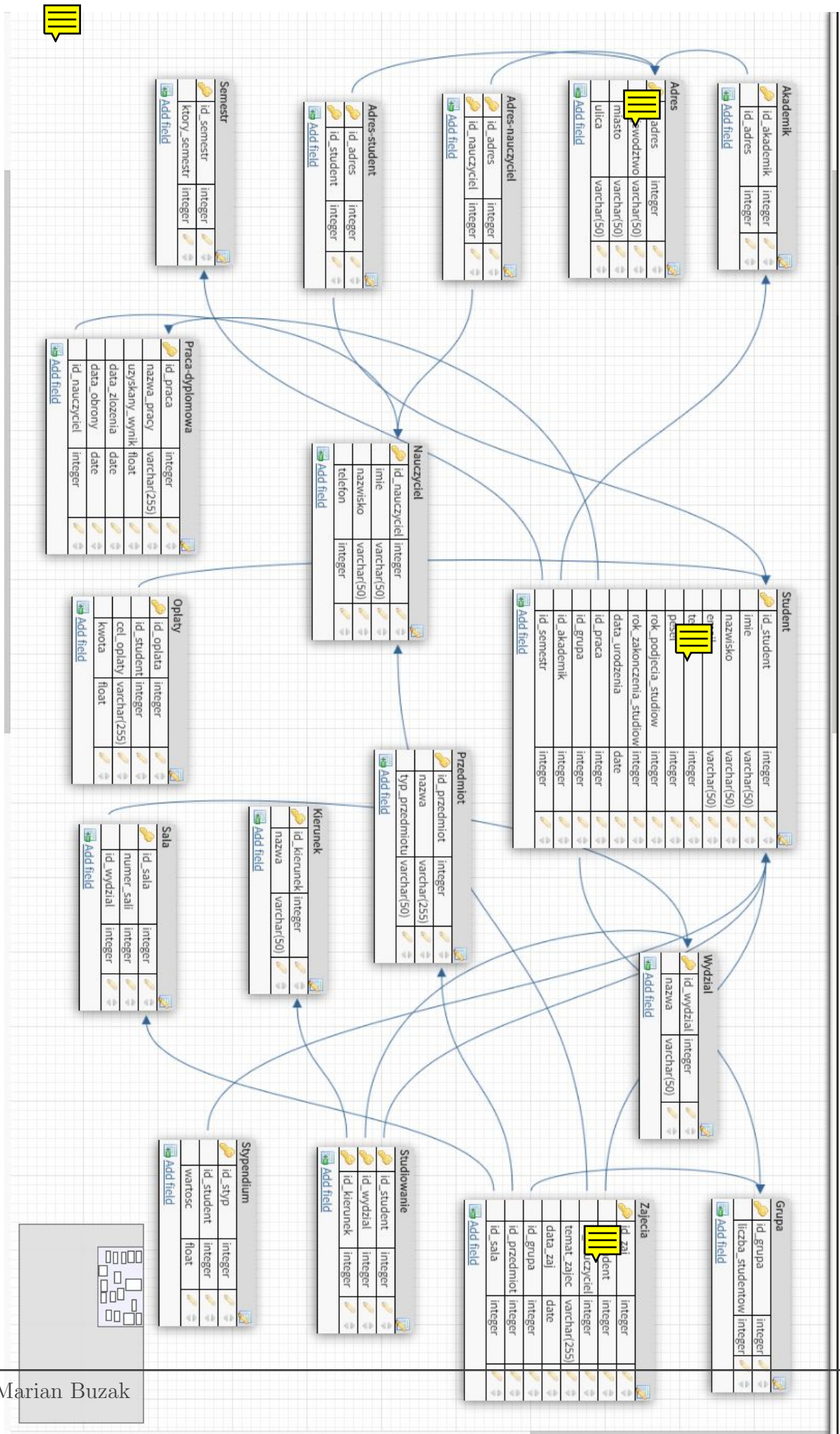
# 3 Przykłady realizacji bazy danych



Przykłady są zrealizowane za pomocą języka MySQL.

## 3.1 Przykłady zawartości najważniejszych tabel

Field	Type	Null	Key	Default	Extra
id_student	int(11)	NO	PRI	NULL	
imie	varchar(50)	NO		NULL	
nazwisko	varchar(50)	NO		NULL	
email	varchar(50)	NO		NULL	
telefon	int(11)	YES		NULL	
pesel	bigint(20)	NO		NULL	
rok_podjecia_studiow	int(11)	NO		NULL	
rok_zakonczenia_studiow	int(11)	YES		NULL	
data_urodzenia	date	NO		NULL	
id_praca	int(11)	YES	MUL	NULL	
id_grupa	int(11)	YES	MUL	NULL	
id_akademik	int(11)	YES	MUL	NULL	
id_semestr	int(11)	YES	MUL	NULL	



Field	Type	Null	Key	Default	Extra
id_zajec	int(11)	NO	PRI	NULL	auto_increment
id_nauczyciel	int(11)	NO	MUL	NULL	
temat_zajec	varchar(255)	NO		NULL	
data_zajec	date	NO		NULL	
id_grupa	int(11)	NO	MUL	NULL	
id_przedmiot	int(11)	NO	MUL	NULL	
id_sala	int(11)	NO	MUL	NULL	

## 3.2 Przykłady kilku zapytań i ich wyników

Zapytanie 1: nazwa oraz typ przedmiotu studenta z indeksem 200001.

```
01 | SELECT nazwa,typ_przedmiotu FROM Przedmiot WHERE id_przedmiot IN(select
    id_przedmiot FROM Zajecia WHERE id_grupa IN(select id_grupa from
    Grupa WHERE id_grupa IN(select id_grupa FROM Student WHERE id_student
    =200001)));
```

nazwa	typ_przedmiotu
matematyka dyskretna	laboratorium
bazy danych	wyklad
matematyka dyskretna	wyklad

Zapytanie 2: adresy studentów, którzy nie mieszkają w akademiku.

```
01 | SELECT * FROM Adres WHERE id_adres IN(select id_adres FROM AdresStudent
    WHERE id_student IN(select id_student FROM Student WHERE id_akademik
    IS NULL));
```

id_adres	wojewodztwo	miasto	ulica
8	pomorskie	Gdansk	Kopernika
9	pomorskie	Gdynia	10 lutego
12	pomorskie	Gdynia	Dolna
15	pomorskie	Gdynia	Traugutta
16	pomorskie	Gdynia	Gieldowa
17	opolskie	Opole	Stryjska

Zapytanie 3: różnica wieku pomiędzy najstarszym i najmłodszym studentem(w latach).

01 | `SELECT (MAX(YEAR(data_urodzenia))-MIN(YEAR(data_urodzenia))) FROM  
Student;`

```
+-----+
| (MAX(YEAR(data_urodzenia))-MIN(YEAR(data_urodzenia))) |
+-----+
|                                                         50 |
+-----+
```



## Literatura

[1] Learn HTML and CSS with w3schools, 2010